Title	組合せ論理回路による有限体上の多項式演算の複雑さ (数理情報科学の基礎理論と応用)
Author(s)	安浦, 寛人; 矢島, 脩三
Citation	数理解析研究所講究録 (1981), 421: 180-192
Issue Date	1981-03
URL	http://hdl.handle.net/2433/102542
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
421巻 1981年 180-192

組合せ論理回路による有限体上の多項式演算の複雑さ

京大 情報工 安浦 寛人
矢島 脩三

1. まえがき

有限体上の多項式演算は，巡回符号をはじめとする誤り制御符号の理論において重要である．ここでは，$GF(2)$ 上の多項式の演算を組合せ論理回路で実現する際の回路の段数および素子数について議論する．$GF(2)$ 上の多項式の演算は，2進整数に対する算術演算における桁上げの効果を無視した場合と見做すこともでき，算術演算の複雑さとも大いに関連がある．ここでは，加算，乗算，平方，除算について，主に段数（計算時間）を中心に議論を進める．

2. 準備

$K=\{0,1\}$ とする．すべての2変数論理関数の集合 $B=\{f \mid f: K^2 \to K\}$ を基底と呼ぶ．基底 B 上の組合せ論理回路 N を以下のように定義する．N はラベルのついた閉路を含まな

い有向グラフで，各節点の入次数は0または2である．入次数0の節点は入力点と呼ばれ，K上の変数$a_1, a_2, \cdots, a_n$または定数0，1のいずれかがラベルとして付けられる．入次数2の節点は計算点と呼ばれ，B中の関数がラベルとして付けられる．

N中の各節点vに対して，その節点の出力となるべき関数$res(v): K^n \to K$を次のように対応させる．

$$res(v) = \begin{cases} v\text{のラベル} & (v\text{が入力点のとき}) \\ g(res(v_1), res(v_2)) & (v\text{がラベル}g\text{を持つ計算点で}v\text{の入力枝は}v_1, v_2\text{から出ているとき}) \end{cases}$$

n変数m出力関数$f: K^n \to K^m$を回路Nが計算するとは，N中に，$f(a_1, a_2, \cdots, a_n) = (res(v_1), res(v_2), \cdots, res(v_m))$となる節点集合$\{v_1, v_2, \cdots, v_m\}$が存在することである．このとき$v_1, v_2, \cdots, v_m$を特に出力点と呼ぶ

組合せ回路Nの段数$depth(N)$とは，入力点から出力点に至るN中のすべての経路の中で最も多くの計算点を含む経路上の計算点の数である．基底B中の各関数を実現する素子の遅延時間が同一であるとすると，段数は回路全体の遅延時間に比例する．N中に含まれる計算点の総数を素子数と呼び，$size(N)$で表わす．論理関数fの計算量$C(f)$および計算時

間 $D(f)$ をそれぞれ f を計算する組合せ回路の素子数および段数の最小値によって定義する.[1]

本稿では, 多項式の演算と論理演算を区別するために, 論理和, 論理積, 排他的論理和をそれぞれ $a \vee b$, $a \wedge b$, $a \oplus b$ によって表わす. $K = \{0, 1\}$ は $\oplus$ を内算法, $\wedge$ を外算法とする有限体 $GF(2)$ である.

3. 加算, 乗算, 平方

3.1 加算

2つの n 次の $GF(2)$ 上の多項式 $A(x)$, $B(x)$ の和を計算するのは, 各次数毎の係数の排他的論理和を求めればよい. すなわち,

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$$

$$B(x) = b_n x^n + b_{n-1} x^{n-1} + \cdots + b_0$$

$$A(x) + B(x) = C(x) = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_0$$

とすると, $c_i = a_i \oplus b_i$ $(i = 0, 1, 2, \cdots, n)$ が成立する. よって加算の計算時間は 1, 計算量は $n + 1$ となる.

3.2 乗算

n 次の多項式 $A(x)$, $B(x)$ の積を

$$A(x)\,B(x) = C(x) = c_{2n} x^{2n} + c_{2n-1} x^{2n-1} + \cdots + c_0$$

とすると, $C(x)$ の各係数 c_k は,

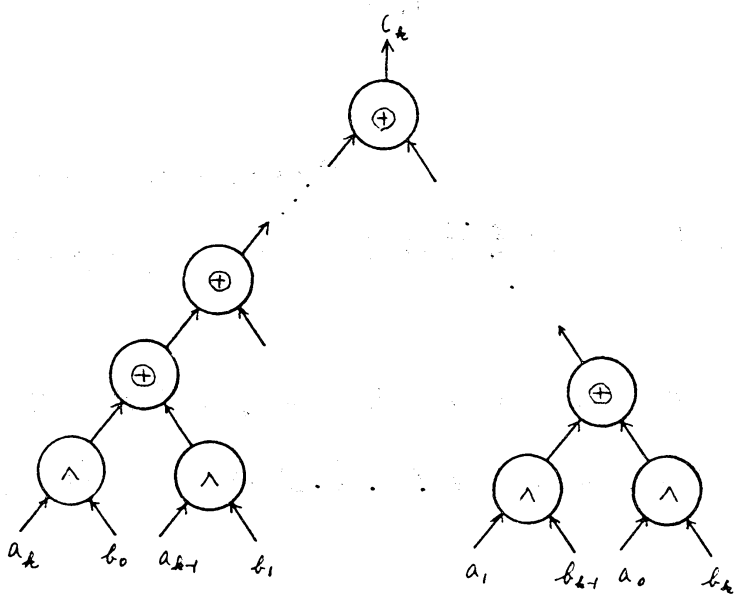

図1．乗算回路 M_k

$$\begin{cases} c_k = a_k \wedge b_0 \oplus a_{k-1} \wedge b_1 \oplus \cdots \oplus a_0 \wedge b_k & (k \leqq n \text{ のとき}) \\ c_k = a_n \wedge b_{k-n} \oplus a_{n-1} \wedge b_{k-n+1} \oplus \cdots \oplus a_{k-n} \wedge b_n & (k > n \text{ のとき}) \end{cases}$$

と表わせる．よって図1のような回路で c_k を計算することができる．$M_0, M_1, \cdots, M_{2n}$ から乗算回路Mを構成すると，

$$depth(M) = \lceil \log_2(n+1) \rceil + 1$$

$$size(M) = 2n^2 + 2n + 1$$

となる．c_n は真に $2n+2$ 変数に依存するから，計算時間の下界は $\lceil \log_2(n+1) \rceil + 1$ となり[2]，この乗算回路Mは段数に関して最適である．すなわち，n 次多項式の乗算の計算時間は，$\lceil \log_2(n+1) \rceil + 1$ となる．また，整数に対する Schönhage - Strassen のアルゴリズム[3] と同様な手法を用いると，計算量は，

$O(n \log n \log\log n)$ となることが示される.

〔命題1〕 GF(2)上のn次多項式の乗算の計算時間は, $\lceil \log_2(n+1) \rceil + 1$, 計算量は $O(n \log n \log\log n)$ である.

さらに, A(x)とB(x)の次数が異なる場合, すなわちA(x)がm次, B(x)がn次のとき乗算の計算時間は, $\lceil \log_2(\mathrm{Min}(n,m)+1) \rceil + 1$ となる.

3.3 平方

$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$ の平方を計算する問題を考える. 一般に, 標数Pの有限体上では, $[A(x)]^P = A(x^P)$ が成立する[4]. よって, GF(2)上での平方の計算は, $P=2$ であるから, $[A(x)]^2 = A(x^2) = a_n x^{2n} + a_{n-1} x^{2n-2} + \cdots + a_1 x^2 + a_0$ となる. すなわち, $[A(x)]^2$ の計算は, 段数, 素子数ともに0でよい.

〔命題2〕 GF(2)上の多項式の平方の計算時間および計算量は0である.

4. 除算

GF(2)上の多項式の除算については, シフトレジスタを用いた簡単な回路構成が知られており, 実用化されているが[4],

組合せ回路による除算についての議論は少ない．

ここでは，n 次多項式 $A(x)$ を m 次多項式 $B(x)$ で割り，商 $Q(x)$ と剰余 $R(x)$ を求める問題を考える．すなわち，

$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$

$B(x) = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_0 \qquad (b_m = 1)$

が与えられたとき，$A(x) = B(x)\,Q(x) + R(x)$ となる $n-m$ 次多項式 $Q(x)$ と $m-1$ 次多項式 $R(x)$ を求める．ここに，

$Q(x) = q_{n-m} x^{n-m} + q_{n-m-1} x^{n-m-1} + \cdots + q_0$

$R(x) = r_{m-1} x^{m-1} + r_{m-2} x^{m-2} + \cdots + r_0$

とする．

〔補題1〕 $A(x)$ を n 次の多項式，$B(x)$ を m 次のモニック多項式 $(b_m = 1)$ とする．$A(x)$ を $B(x)$ で割ったときの商を $Q(x)$，x^n を $B(x)$ で割ったときの商を $P(x)$ とすると，

$$A(x)\,P(x) = Q(x)\,x^n + T(x)$$

が成立する．ここに $T(x)$ は $n-1$ 次以下の多項式である．

（証明） $x^n = B(x)\,P(x) + S(x)$，$A(x) = B(x)\,Q(x) + R(x)$ と表わせる．ここに $S(x)$，$R(x)$ はともに $m-1$ 次の多項式である．そこで，

$$A(x)\,P(x) = (B(x)\,Q(x) + R(x))\,P(x)$$

$$= B(x)\,P(x)\,Q(x) + R(x)\,P(x)$$

$$= (x^n + S(x))\,Q(x) + R(x)\,P(x)$$

$$= Q(x)\,x^n + S(x)\,Q(x) + R(x)\,P(x)$$

ここで，$P(x)$ と $Q(x)$ はともに $n-m$ 次の多項式であり，$S(x)$ と $R(x)$ が $m-1$ 次以下なので，$S(x)\,Q(x) + R(x)\,P(x)$ は $n-1$ 次以下の多項式である．（証明終）

この $P(x)$ を $B(x)$ の逆多項式と呼ぶ．補題1より，$B(x)$ の逆多項式 $P(x)$ をあらかじめ作っておけば，$Q(x)$ は $A(x)$ と $P(x)$ の積の高次の項として得られる．しかも，$Q(x)$ は $A(x)P(x)$ の上位 $n-m$ 項であるから，乗算は上位 $n-m$ 項だけを計算するだけでよい．

以上のような考察より，図2のような回路で除算を行なうことができる．まず逆多項式回路で $B(x)$ の逆多項式 $P(x)$ を作り，$A(x)$ と $P(x)$ の積を求めることによって商 $Q(x)$ を得る．剰余 $R(x)$ は，$R(x) = A(x) - B(x)\,Q(x)$ に従って算出する．

逆多項式回路は，$B(x)$ を入力として $P(x)$ を出力する．この回

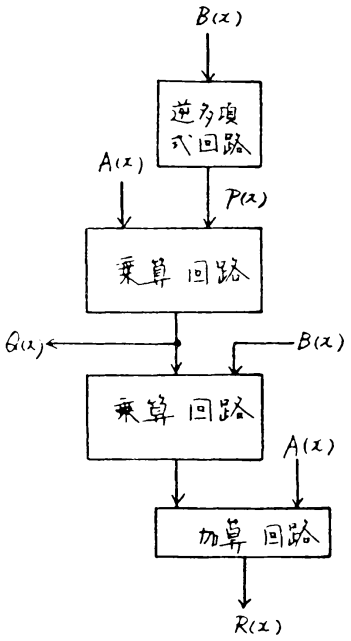

図2　除算回路

路に用いるアルゴリズムは，次のようなものである．

〔アルゴリズムP〕

入力：m次多項式 $B(x)$

出力：$B(x)$の逆多項式 $P(x)$ （x^nを$B(x)$で割った商）

方法：$k=\lceil \log_2(n-m+1) \rceil$ とする．

ステップ1　$(B(x))^{2^i}$ を $x^{2^i m-n+m}$ で割った商 $V_i(x)$ を $i=0,1,\cdots,k-1$ について求める．

ステップ2　$\prod_{i=0}^{k-1} V_i(x)$ を $x^{2^k m-n}$ で割った商を $P(x)$ とする．

アルゴリズムPが，正しく$B(x)$の逆多項式$P(x)$を計算することを示すため，次の補題を示す．

〔補題2〕　$F(x)$をn_0次多項式，$D(x)$，$E(x)$をそれぞれn_1次，n_2次の多項式とする．$F(x)$が$D(x)$と$E(x)$の積であるとき，$F(x)$をx^{m_0}（但し，$2m_0>n_0$）で割った商$F'(x)$は，

$$D'(x)\,E'(x)=F'(x)\,x^{n_0-m_0}+G(x)$$

をみたす．但し，$D'(x)$，$E'(x)$はそれぞれ$D(x)$，$E(x)$を$x^{n_1-(n_0-m_0)}$，$x^{n_2-(n_0-m_0)}$で割った商であり，$G(x)$は高々n_0-m_0-1次の多項式である．

（証明）　$F(x)$，$E(x)$，$D(x)$はそれぞれ

$$F(x) = F'(x)\,x^{m_0} + F''(x) \qquad (F''(x) \text{ は } m_0-1 \text{ 次以下})$$

$$D(x) = D'(x)\,x^{n_1-(n_0-m_0)} + D''(x) \qquad (D''(x) \text{ は } n_1-(n_0-m_0)-1 \text{ 次以下})$$

$$E(x) = E'(x)\,x^{n_2-(n_0-m_0)} + E''(x) \qquad (E''(x) \text{ は } n_2-(n_0-m_0)-1 \text{ 次以下})$$

とかける．よって，

$$D(x)\,E(x) = E'(x)\,D'(x)\,x^{n_1+n_2-2(n_0-m_0)} + D''(x)\,E'(x)\,x^{n_2-(n_0-m_0)} + D'(x)\,E''(x)\,x^{n_1-(n_0-m_0)} + D''(x)\,E''(x)$$

右辺の各項の最高次数は，$n_0 = n_1 + n_2$ であることを考えると，各々，n_0 次，m_0-1 次，m_0-1 次，$2m_0-n_0-1$ 次となる．$n_0 > m_0$ であるから，第2項以下はすべて m_0-1 次以下となる．よって，$F'(x)$ は $D'(x)E'(x)$ の上位 n_0-m_0 次分の係数と一致する．(証明終)

アルゴリズム P が正しく $P(x)$ を計算することを示す．3.3 に述べた平方の性質より，

$$(B(x))^{2^k} = B(x^{2^k}) = x^{2^k\cdot m} + U(x) \qquad (U(x) \text{ は } 2^k m - 2^k \text{ 次以下})$$

である．今，

$$x^{2^k\cdot m} = x^n \cdot x^{2^k m-n} = (B(x)P(x) + S(x))\,x^{2^k\cdot m-n} = B(x)P(x)\,x^{2^k m-n} + S(x)\,x^{2^k m-n}$$

と書ける．

$$S(x)\,x^{2^k\cdot m-n} = B(x)\,W(x) + X(x), \qquad (X(x) \text{ は } m-1 \text{ 次以下})$$

$$U(x) = B(x)\,Y(x) + Z(x) \qquad (Z(x) \text{ は } m-1 \text{ 次以下})$$

としたとき，

$$(B(x))^{2^k} = x^{2^k \cdot m} + U(x)$$

$$= B(x)\left(P(x)x^{2^k \cdot m - n} + W(x) + Y(x)\right) + X(x) + Z(x)$$

と書けるが，$(B(x))^{2^k}$ は $B(x)$ でわりきれるから，$m-1$ 次以下の項 $X(x)+Z(x)$ は 0 となる．よって，

$$(B(x))^{2^k-1} = P(x)x^{2^k \cdot m - n} + W(x) + Y(x)$$

となる．ここに $W(x)$ は $2^k \cdot m - n - 1$ 次以下であり，$Y(x)$ は $2^k m - 2^k - m$ 次以下である．k の定義より，$n - m + 1 \leqq 2^k$ であるから，$Y(x)$ も $2^k \cdot m - n - 1$ 次以下となる．すなわち，$(B(x))^{2^k-1}$ の上位 $n-m+1$ 項の係数は $P(x)$ と一致する．よって，

$$(B(x))^{2^k-1} = \prod_{i=0}^{k-1} (B(x))^{2^i}$$

を計算すればよい．さらに，補題2より，各 $(B(x))^{2^i}$ の上位 $n-m+1$ 項の積だけを計算すればよいので，アルゴリズム P のように $V_i(x)$ を作ってそれらの積を作ればよい．以上の議論より，アルゴリズム P が $P(x)$ を正しく計算することが示された．

各 $V_i(x)$ は $B(x)$ から直接次のようにして作れる．

$V_0(x) = B(x)$ の上位 $n-m+1$ 項

$V_{i-1}(x) = v_{n-m}x^{n-m} + v_{n-m-1}x^{n-m-1} + \cdots + v_0$

$V_i(x) = v'_{n-m}x^{n-m} + v'_{n-m-1}x^{n-m-1} + \cdots + v'_0$

とすると，$v'_{n-m-2j} = v_{n-m-j}$，$v'_{n-m-2j-1} = 0$ $\left(j = 0, 1, \cdots, \frac{n-m}{2}\right)$．すなわち $V_i(x)$ は $B(x)$ から素子数 0 で作ることができる．ア

ルゴリズムIに基づく逆多項式回路は，図3のように構成できる．各乗算回路は $n-m$ 次の乗算を行なえばよく，$\lceil \log_2(n-m+1) \rceil + 1$ 段で構成できる．乗算回路からなる木の深さは，

$\lceil \log_2 k \rceil =$

$\lceil \log_2 (\lceil \log_2 (n-m+1) \rceil) \rceil$

であり，回路全体の段数は，

$(\lceil \log_2 (n-m+1) \rceil + 1) \lceil \log_2 (\lceil \log_2 (n-m+1) \rceil) \rceil$

となる．以上の議論と，命題1，命題2より次の定理を得る．

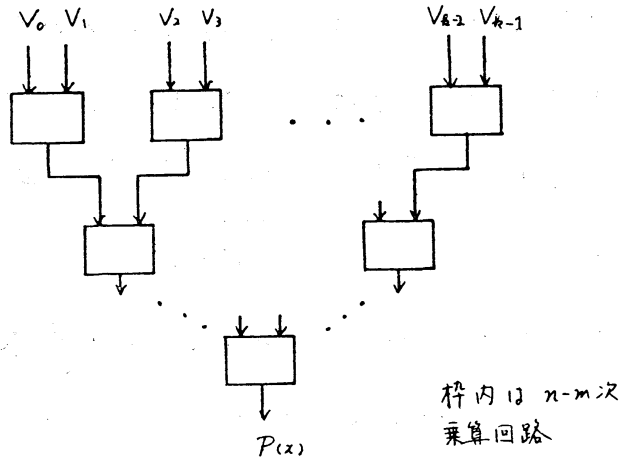

図3 逆多項式回路

〔定理〕 GF(2) 上の n 次多項式を n 次の多項式で割る除算は，計算時間 $O(\log n \log\log n)$，計算量 $O(n \log n)^2 \log\log n)$ である．

逆多項式回路に入力される $V_0, V_1, \cdots, V_{k-1}$ は，それぞれ係数が0でない項は，$n-m+1$ 項，$(n-m+1)/2$ 項，$(n-m+1)/4$ 項，…，1項以下である．従って，乗算回路は係数が

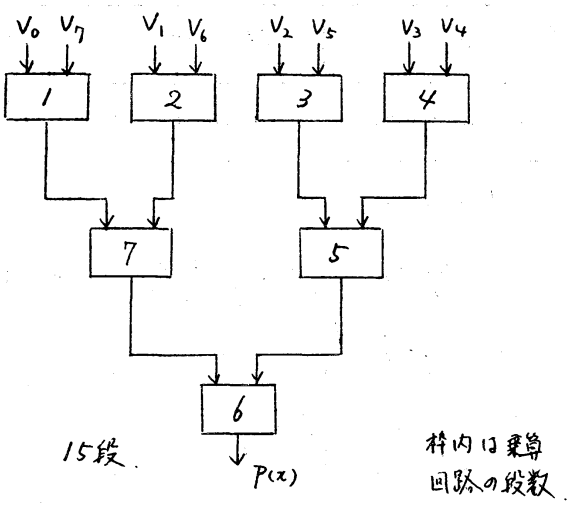

図4　逆多項式回路の段数削減

0でない部分の計算だけ行なえばよく，段数，素子数を削減できる．例えば，$k=8$ の場合，図3の形でこのようなことを考慮して回路を作成すると，17段の回路となるが，さらに，図4のように，V_i の入力する順番を適当に変えることにより，15段の回路が構成できる．このような，回路の最適化は，実用上重要と考えられる．

5．あとがき

GF(2)上の多項式の加算，乗算，平方，除算を組合せ論理回路によって計算するときの素子数と段数について考察した．ここで示した結果は，容易に一般の有限体GF(8)上へ拡張できる．

最後に，2進数の算術演算とGF(2)上の多項式演算の複雑さを表1に比較しておく．GF(2)上の多項式演算は，桁上げの効果を無視した時の算術演算と同じと考えられる．表1に

	算術演算		多項式演算	
	計算量	計算時間	計算量	計算時間
加(減)算	$\theta(n)$	$\theta(\log n)$	n	1
乗算	$O(n\log n\log\log n)$	$\theta(\log n)$	$O(n\log n\log\log n)$	$\lfloor\log n\rfloor+1$
平方	$O(n\log n\log\log n)$	$\theta(\log n)$	0	0
除算	$O(n\log n\log\log n)$	$O((\log n)^2)$	$O(n(\log n)^2\log\log n)$	$O(\log n\log\log n)$

表1. 算術演算と多項式演算の複雑さ

示すように，桁上げ効果の分だけ多項式演算の複雑さが低くなっていると思われる．

謝辞．御討論頂いた本学上林弥彦助教授をはじめとする矢島研究室の諸氏に感謝いたします．

文献．

(1) 安浦"論理関数の複雑さの理論とその高速論理回路の構成法への応用"，情報処理論文誌，Vol. 21, No. 4, 1980年7月．

(2) 安浦，矢島"論理関数を実現するのに必要な論理回路の段数について"，信学論 Vol. 62, No. 9, 1979年9月．

(3) Aho, Hopcroft, Ulman "Design and Analysis of Computer Algorithms", Addison-Wesley 1974.

(4) 宮川，岩垂，今井"符号理論"，昭晃堂，1973．